CS409 mp2 (part1): Testing via Input Domain Modeling

(Total: 20 points, 5 points bonus)

Deadline: November 13, 11.59pm

There are two parts for this MP and each part has different invitation link. In this part (Invitation links: https://classroom.github.com/a/Po6XtyWE), each student will apply Input Domain Modeling learned in class for an open-source project, Joda-time. Note that you need to submit README.md and JUnit test classes for this MP part 1 (For part 1, you need to **overwrite the README.md in Joda-Time**).

Find the following method in Joda-Time:

```
/**
        * Returns a new duration with this length plus that specified multiplied by the scalar.
        * This instance is immutable and is not altered.
        * <p>
        * If the addition is zero, this instance is returned.
        *
        * @param durationToAdd the duration to add to this one
        * @param scalar the amount of times to add, such as -1 to subtract once
        * @return the new duration instance
        */
        public Duration withDurationAdded(long durationToAdd, int scalar)
        { if (durationToAdd == 0 || scalar == 0) { return this;

        }
        long add = FieldUtils.safeMultiply(durationToAdd, scalar);
        long duration = FieldUtils.safeAdd(getMillis(), add);
        return new Duration(duration);
        }
```

***Do not forget to list the names and student ids of the members in your group. Do not foget to include the name and link of your selected app.**

a) What are the parameters of this public method? Remember that parameters includes arguments to the method and the state variables. (2 points)

b) Identify the characteristic according to Interface-based Input Domain Modeling. *Make sure that your partitions are complete and disjoint. (2 points for identify characteristic)*

   i. What are the partitions according to this characteristic? (2 points for identify partitions)

   ii. What are the test inputs that fulfills Each Choice Coverage? (2 points)

   iii. Write JUnit tests for the test inputs in ii. (3 points)

c) Identify the characteristic according to Functionality-based Input Domain Modeling. *Make sure that your partitions are complete and disjoint. (2 points for identify characteristic)*

i. *What are the partitions according to this characteristic? (2 points for identify partitions)*
ii. *What are the test inputs that fulfills Each Choice Coverage? (2 points)*
iii. *Write JUnit tests for the test inputs in ii. (3 points)*

2. You can do the following to get bonus points (5 points):
   a) If you find a bug, post the bug that you find in GitHub by posting it at (Read the bug reports requirements below and the example bug report at the link below before posting): https://github.com/orgs/cs409-software-testing2020/teams/allstudents/discussions/2

Bug reports requirements (Java programs)

A good bug report should include the following:

- Detailed Inputs: If the bug requires a specific image/file to trigger, this should be included.
- JUnit tests: Include the JUnit tests for reproducing the faults
- Version tested: The version of the program that you have tested should be included
- Patch: If you could issue a pull request for patching the bug, then you should include this in your bug report.
- Duplicate: Has this bug report been previous posted?

CS409 mp2 (part2): Testing via Input Domain Modeling

(Total: 20 points, 5 points bonus)

Deadline: Deadline: November 13, 11.59pm

In this part (Invitation links: https://classroom.github.com/a/TrnLrnQP),

each team member (**individual assignment**) will apply Input Domain

Modeling learned in class for their selected app.

Bug reports requirements (Android apps)

A good bug report should include the following:

- Steps to reproduce: The steps (clicks, touches, etc) that leads to crash should be included
- Detailed Inputs: If the bug requires a specific image/file to trigger, this should be included
- Stacktrace: If the fault leads to a crash, the stacktrace information should be included
- Device Information: The information on the device used for testing, including Android versions and Phone Model Used should be included.

- Version tested: The version of the app that you have tested should be included
- Patch: If you could issue a pull request for patching the bug, then you should include this in your bug report.
- Duplicate: Has this bug report been previous posted?

***Do not forget to list the names and student ids of the members in your group. Do not forget to include the name and link of your selected app.**

3. Select a public method in your app. Add this public method in the README.md by quoting the source code.
    a) What are the parameters of this public method? Remember that parameters includes arguments to the method and the state variables. (2 points)

b) Identify the characteristic according to Interface-based Input Domain Modeling. *Make sure that your partitions are complete and disjoint. (2 points for identify characteristic)*

   i. *What are the partitions according to this characteristic? (2 points for identify partitions)*
   ii. *What are the test inputs that fulfills Each Choice Coverage? (2 points)*
   iii. *Write JUnit tests for the test inputs in ii. (3 points)*

c) Identify the characteristic according to Functionality-based Input Domain Modeling. *Make sure that your partitions are complete and disjoint. (2 points for identify characteristic)*

   i. *What are the partitions according to this characteristic? (2 points for identify partitions)*
   ii. *What are the test inputs that fulfills Each Choice Coverage? (2 points)*
   iii. *Write JUnit tests for the test inputs in ii. (3 points)*

4. You can do either of the following to get bonus points (5 points):

   a) If you find a bug, post the bug that you find in GitHub by posting in your team discussion. Read the https://github.com/orgs/cs409-software-testing2020/teams/allstudents/discussions/2 for example bug report. If all your team members checks that your bug reports fulfilled the bug reports requirement, post it at: https://github.com/orgs/cs409-software-testing2020/teams/allstudents/discussions/

   b) If you read a GitHub issue in the GitHub Discussion (Team or All-Students discussion) within your group and could use the information in the GitHub issue for finding a "similar" bug in your selected app, comment in GitHub by posting at: https://github.com/orgs/cs409-software-testing2020/teams/allstudents/discussions/. *You need to add a bug reports (check the bug report requirement) and add a comment on "What information is useful for sharing this bug?"*