A1: Code Review of The Fish Tank

The Fish Tank: You're Hired

A company hired a programmer to implement their **specification for a fish tank (https://q.utoronto.ca/courses/68685/pages/fish-tank-specification)** simulator. That programmer has moved on to another job, and you have been hired to replace them. The system is partially completed, and your first task is to do a code review. There are three parts to this:

1. Learning enough Git to get a copy of the code into IntelliJ.
2. Finding and analyzing code smells.
3. Fixing the functionality of the program so that it matches the specification. (i.e., Make it do what it is supposed to do.)

Git

We provide the starter code in a remote repository, which is like a database. You need to clone the database to get a copy onto your computer. When you make changes, you add and commit them to your local copy, and then you push the changes to the remote repository.

Please do not add any new files to the Git repo until you are able to run the code. You will be asked by IntelliJ about xml files and so on. Always, always select No.

Here's how to clone:

1. Log into MarkUs: **https://markus.teach.cs.toronto.edu/csc207-2019-01 (https://markus.teach.cs.toronto.edu/csc207-2019-01)** .
2. Go to Assignment 1. You'll see the repository URL on the right-hand side of the screen. Click on it and also copy it.
3. Open IntelliJ. (You can do this on the lab computers by opening a Terminal and typing "idea".)
4. Close all your projects. From the **Welcome to IntelliJ IDEA** window, select **Check out from Version Control**.
5. Paste the URL you found on the Assignment 1 page and click Test to make sure you are using the correct URL.
6. Follow the prompts. **DO NOT ADD ANY FILES TO THE GIT REPO WHILE SETTING UP.** Select "no" whenever you are asked.

Running the code

You now have a local copy of the repo. Right now, all the files are inside nested folders. To get to them, you can use the commands:

"**ls**" to list the files in the current directory "**cd foldername**" to move into one of those directories (replace **foldername** with the name of the folder into which you want to move)

"**cd ..**" to get back out of that folder.

Currently your code should be in **Assignment1/fishtank/src/fishtank/** and your **codesmells.md** file should be in **Assignment1/** folder.

Back in IntelliJ, right-click on **Assignment1 -> fishtank -> src -> fishtank -> FishTank.java** and select **Run**.

If **Run** is not an option, you may need to tell IntelliJ that it's a Java project and you need to tell it that the **src** directory is the root of the Java code. If you are in this situation, then do the following:

1. Open up the Project Structure.
2. Set the SDK to 1.8.
3. Set the Project language level to 8.
4. Right-click on **src** and select **Mark Directory As -> Sources Root**

Finding code smells

1. One of the files is called "codesmells.md". Open it and read it.
2. In a browser, open https://sourcemaking.com/refactoring/smells (https://sourcemaking.com/refactoring/smells) .
3. Find a code smell in the code and describe it using the template.
4. Now it's time to add, commit, and push your change.
   1. Right-click on the codesmells.md file.
   2. Select **Git -> add**, then **Git -> commit**. You'll be prompted to type a "commit message". Type something short and clear, like "Described Change Preventer code smell." (The **Git** option may say **VCS** instead, for Version Control System.)
   3. Now your local copy has recorded your changes. To push them back to the remote repository (thereby submitted the assignment!), choose **Git -> push**.
5. Find four more different code smells from at least **three** categories (Bloater, etc.), and report on them using the template. **After each one**, add, commit, and push. **We will check to make sure you have committed several times, rather than all at once.**
6. When describing a code smell, assume that you have not fixed any of the previous smells. You are always describing the code the way it was when it was given to you. If the same fix corrects more than one code smell, you still have to describe it twice and explain, independently, why it fixes each.
7. Read the specification (https://q.utoronto.ca/courses/68685/pages/fish-tank-specification) and compare it to the code. Are there any problems with the functionality of the code? If so, find them and fix them. Add, commit, and push the final version of all of your files.

What gets marked?

Each time you push, you should submit a new version of the .md file. The last one pushed before the deadline is the version we will mark. Similarly, the most recent version of your .java files will also be marked. For A1, you should push whenever you think you're done describing a code smell. **You can edit your smells later with no penalty. Just push a new version.**