CS3318 | Advanced Programming with Java

Individual Assignment:Test Driven Development

Content

- Introduction, Deliverables & Outcomes
- Benefits of Test Driven Development

Report Content

- What is a Colour Palette and can we paint with it?
- The Great Maven Challenge (an unofficial part 3)
- Creating an Array of Colours!
- To TDD or not to TDD - *that is the question?*
- Learning Experiences and Goodbyes

Introduction, Deliverables & Outcomes

Test Driven Development (TDD) is a software development approach in which test cases specify and validate the functionality of the code. The TDD approach is described as a cycle of Red-Green-Refactor. Run a test, see it fail (red), implement the simplest code to make the test pass (green), and then refactor the code so your test stays green and your code is improved.

During this assignment, following a TDD approach requires creating the test class, creating the test methods, and then creating empty implementations of the code that will eventually become production code.

I hope to develop an application that will make use of these TDD principles. I will also use an IDE called IntelliJ to help me automate a lot of these processes. Alongside IntelliJ, I will make use of the Maven Support Framework and J-Unit to assist with the application development and testing procedures.

Deliverables

- Source code (Java file)
- Test code (Java file)
- Archive of my Git Repository (link to GitHub repository)
- Git Log (txt file)
- Report (PDF file)

Outcomes

By the end of this assignment, I should be familiar with basic TDD principles and capable of applying them in my own future work. I should also understand the importance of such an approach in the coding industry and learn to make use of it in the workplace.

This type of approach would have been handy in assisting us (students) in the previous group assignment as it would have allowed for writing up tests to help remove a lot of the errors and to optimise the code of the Qr-code scanning API quicker.

Benefits of Test Driven Development

Test-Driven Development (TDD) is a software development methodology that emphasises writing tests before writing the actual code. The benefits of this approach versus standard methods can be summarised by:

- Improved Code Quality
- Faster Debugging
- Faster Development (modular, consistent development)
- Documentation (tests explain the code)
- Fearless Refactoring (can easily update/modenrise code)
- Continuous Integration (automated tests for newly committed code)
- Collaboration (tests provide a guideline)
- Client Confidence (can tell at a glance that)
- Better Design
- Reduction of Technical Debt (reworking code & maintenance)

In summary, Test-Driven Development is a valuable practice that contributes to improved code quality & functionality, faster development cycles, and increased confidence in the reliability of software that we write.

Coding Report

What is a Colour Palette and can we paint with it?

In this assignment we were tasked with developing a **ColourTable** class that can represent a palette of RGB colours. As images are encoded using a colour palette, colour information is not directly carried by the image pixel data, but is stored in a colour lookup table. In my case, this will be stored via a Java ArrayList.

In the ArrayList, every element shall represent a colour inside of the image and the index associated with it is its position in the image.

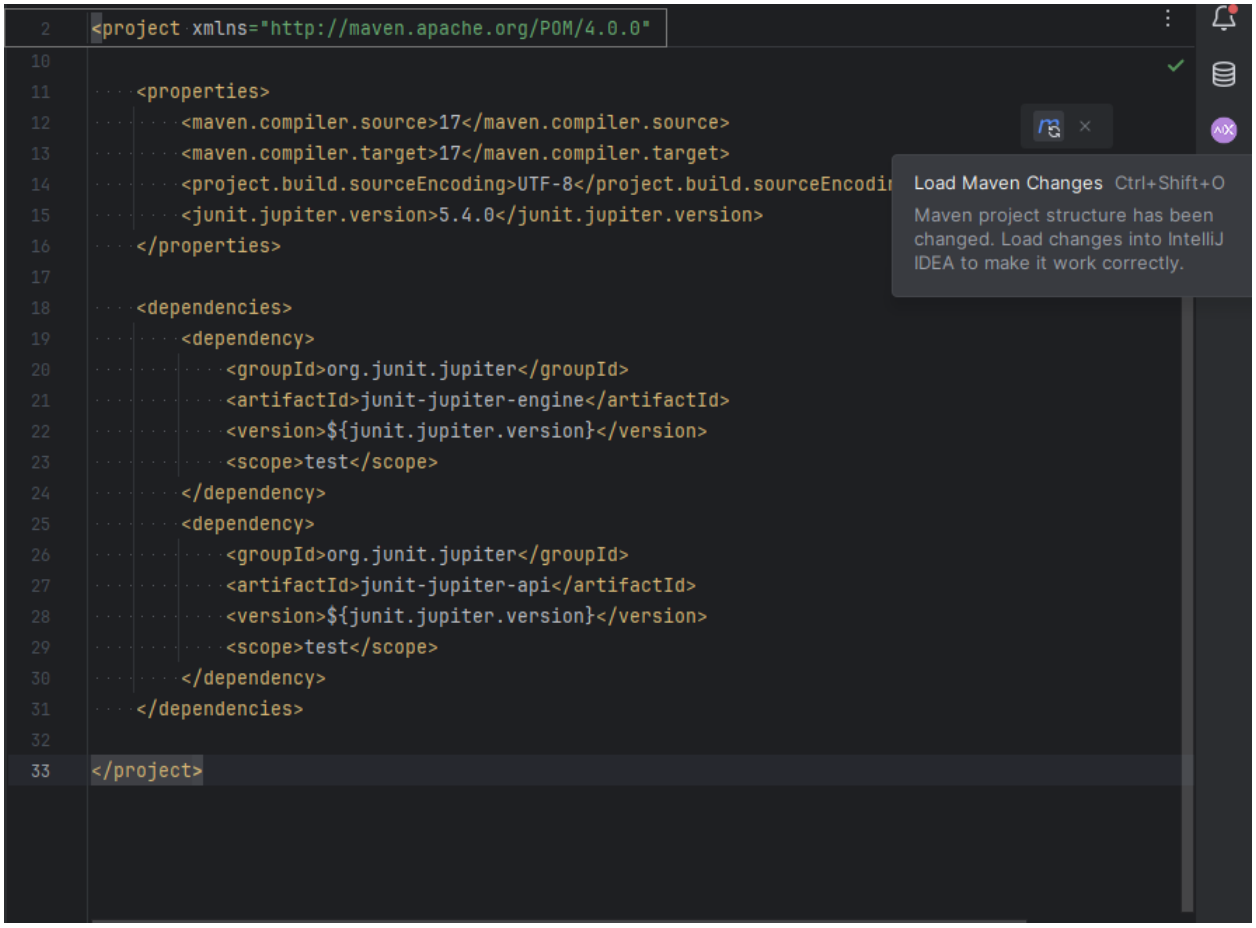
Note that in this assignment, we were not tasked with decoding the image, just creating a means of storing the associated data, so the class created might require a bit of tweaking to work correctly!

The Great Maven Challenge (an unofficial part 3)

For this assignment, I too had to make use of the Maven Framework Support in my Java project. But there was a twist to it! I had to use the J-Unit testing suite!

Initially, adding this to the project proved to be quite difficult as the keywords were not matching up correctly or there would be a missing line, but, with a big of research on the internet, I found a means by which to add the J-Unit Jupiter Engine and API to my Maven Pom file.

With that out of the way, I could begin writing my Unit tests!




```
<project xmlns="http://maven.apache.org/POM/4.0.0"

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncodi
        <junit.jupiter.version>5.4.0</junit.jupiter.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-engine</artifactId>
            <version>${junit.jupiter.version}</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-api</artifactId>
            <version>${junit.jupiter.version}</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

</project>
```

Creating an Array of Colours!

After the initial (Mave-ee) set-up was out of the way, I proceeded to write my Unit tests, executing them, encountering errors, writing program code to nullify said errors and looping back to writing tests again.

For my implementation of the Palette of Colours, I used an ArrayList made up of Strings of RGB codes. The array was limited to be a min size of 2, with a max of 1024 and had to be a Power of 2. There were checkers for all 3 of these cases in-place. In case of being triggered, an Error would be thrown to the User.

The Array itself contained Strings. For my system I used the Hexadecimal system of colours, such that a single colour could be represented with a series of 6 numbers (or letters). Bellow are some examples of these 6 digit codes:

For adding to the Array, getting its size and printing its elements I made the required tests and wrote the methods.

The Array allowed the addition of duplicate entries, but would still alert the user when a duplicate entry had been made.

The Array also had a counter in-place, such that if you went over this limit, it would no-longer add the inputted colour to the Array and would throw you an Error.

To TDD or not to TDD - *that is the question?*

Test Driven Development was a new approach to coding that I had heard about before, but, personally, had never tried! It was a novel and fast-paced experience for me as; what shocked me the most was that writing the tests first actually sped up my work on the task at hand. I could understand what was needed of me to a greater degree, and whenever I got stuck, the Googling of workarounds online was very precise and always yielded a positive answer to me.

In terms of detriments or negatives associated with TDD and don't see any major issues with it. But it does seem to be a very roboticized approach to coding. So much so, that for the majority of programists who think logically - this is not a problem, but if you were to grab someone outside of the programming world and show them this approach they would get a very skewed understanding of what code is and why people need to spend time in college learning it as a degree.

Learning Experiences and Goodbyes

In terms of learning experiences associated with this assignment, I would firstly like to thank our Professor, John O' Mullane for the opportunity to learn one of my favourite programming languages in greater depth and to provide me with new tools to push my understanding of the programming world further and closer to what those in the industry world have! Thank you for that! I will definitely make the best use of this in my future (*looking at you Work Placement!*).

Over the course of this assignment, carefully reading the Project Brief had been of utmost importance, in particular, the section about what the role of the **Colourtable** class would be. Understanding what was needed of you was very important for this assignment and served to me as experience for what working independently on a project feels like.

Having the majority of the brief left to open interpretation, allowed for my creativity to create and fill in the blanks accordingly, leading to a more personal feel to this project. This has taught me that assignments should not be viewed as mere homework, but as opportunities to showcase who you are, what you know and what you value in your work!

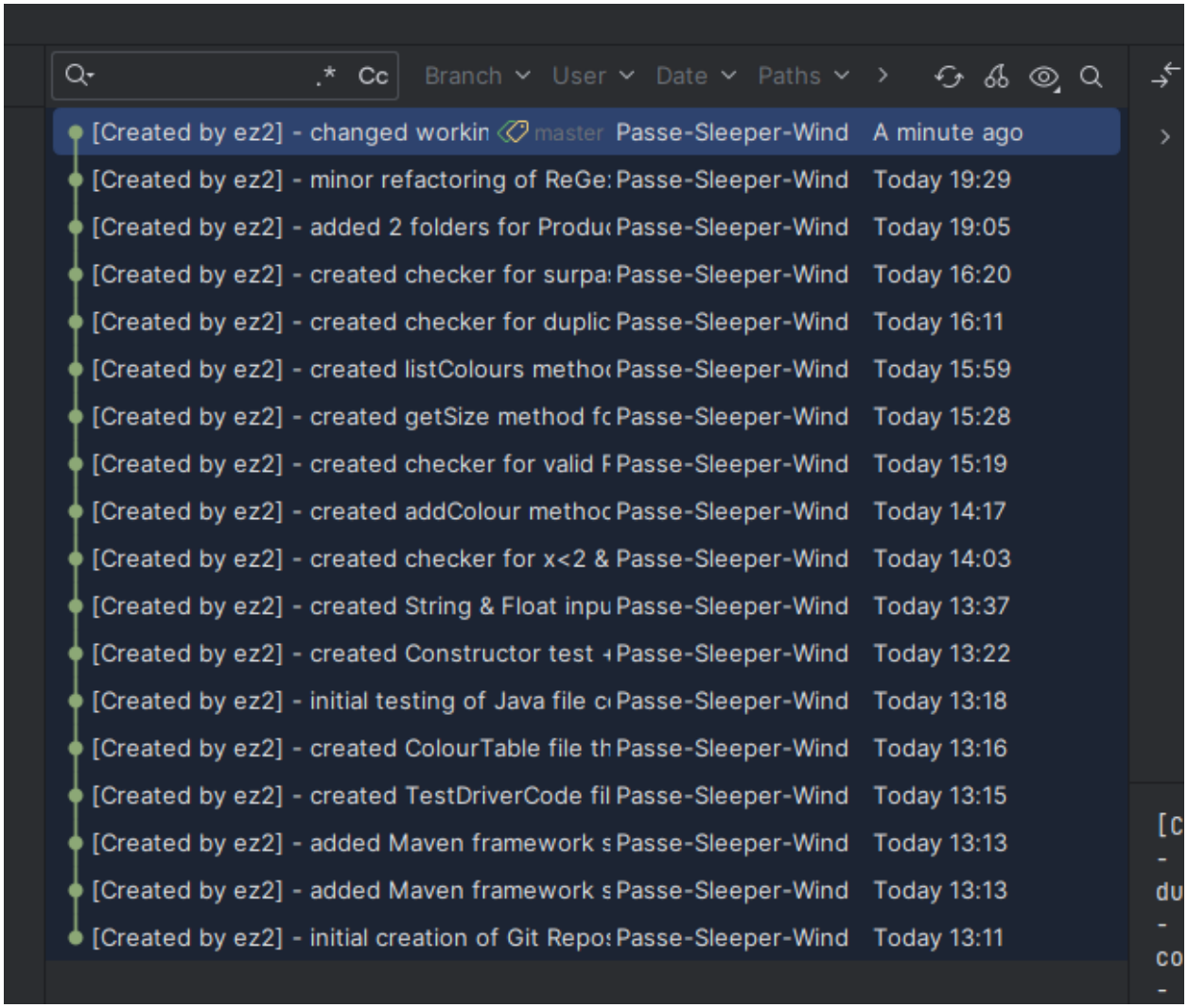
Thank you again for teaching us for this Semester!

I have attached the source code and tester code to this submission.
There is also a complete repository that can be found here. If that doesn't work, follow this link:

https://github.com/Passe-Sleeper/UNI-Public/tree/main/Year3/CS3318

Screenshot of Git Log (I have also added a txt file with all the commits) below.

Note: though I did use the inbuilt Git support in IntelliJ, I afterwards uploaded everything to my GitGub account!


Branch
User
Date
Paths
[Created by ez2] - changed workin master Passe-Sleeper-Wind A minute ago
[Created by ez2] - minor refactoring of ReGe: Passe-Sleeper-Wind Today 19:29
[Created by ez2] - added 2 folders for Produ Passe-Sleeper-Wind Today 19:05
[Created by ez2] - created checker for surpa Passe-Sleeper-Wind Today 16:20
[Created by ez2] - created checker for duplic Passe-Sleeper-Wind Today 16:11
[Created by ez2] - created listColours metho Passe-Sleeper-Wind Today 15:59
[Created by ez2] - created getSize method fo Passe-Sleeper-Wind Today 15:28
[Created by ez2] - created checker for valid F Passe-Sleeper-Wind Today 15:19
[Created by ez2] - created addColour metho Passe-Sleeper-Wind Today 14:17
[Created by ez2] - created checker for x<2 & Passe-Sleeper-Wind Today 14:03
[Created by ez2] - created String & Float inpu Passe-Sleeper-Wind Today 13:37
[Created by ez2] - created Constructor test Passe-Sleeper-Wind Today 13:22
[Created by ez2] - initial testing of Java file c Passe-Sleeper-Wind Today 13:18
[Created by ez2] - created ColourTable file th Passe-Sleeper-Wind Today 13:16
[Created by ez2] - created TestDriverCode fil Passe-Sleeper-Wind Today 13:15
[Created by ez2] - added Maven framework s Passe-Sleeper-Wind Today 13:13
[Created by ez2] - added Maven framework s Passe-Sleeper-Wind Today 13:13
[Created by ez2] - initial creation of Git Repo Passe-Sleeper-Wind Today 13:11