COMP5911M Advanced Software Engineering

Coursework 1

This assignment is based on material from the lectures and worksheets on refactoring. There are 3 separate tasks, with a total of 30 marks available. Each task has multiple questions associated with it. You should submit your answers to these questions using Gradescope.

This assignment is worth 15% of your overall grade.

Task 1

Swing is a framework provided in the Java API to support the development of desktop graphical user interfaces[1]. Swing includes the class `JTable`, a GUI component that represents data in tabular form. To view the API documentation for this class, visit

`https://docs.oracle.com/en/java/javase/17/docs/api/`

Then enter `JTable` in the Search box.

1. What is the main 'code smell' evident from studying the API documentation for `JTable`? Explain your reasoning. **[2 marks]**
2. Describe briefly how you could refactor this class to make it easier to work with. You do not need to go into great detail here, but you should identify any of the standard refactorings that might be useful, and indicate how they would be used here. Use the Catalog of Refactoring to guide you:

   `https://refactoring.guru/refactoring/catalog`

   [4 marks]

Task 2

The Java API also provides a class named `BufferedImage`, to support the creation of image processing applications in Java. To view the API documentation for this class, visit

`https://docs.oracle.com/en/java/javase/17/docs/api/`

Then enter `BufferedImage` in the Search box.

Study the API documentation for the methods of this class, paying particular attention to the `getRGB`, `getSubimage` and `setRGB` methods. Note that you should focus on the API here; the implementation of the methods is irrelevant.

1. What 'code smell' is evident in the API of these methods? Explain your reasoning. **[3 marks]**
2. Identify one of the standard refactorings that would improve these methods, in particular. (Use the Catalog of Refactoring to help you choose.) Explain briefly why your chosen refactoring improves the class and the code that uses it. **[3 marks]**
3. Write some fragments of Java source code that illustrate the improvements that result from your chosen refactoring. **You do NOT need to submit complete, syntactically correct Java here**, but make sure that your code fragments illustrate both the changes made to the class and the changes in how the class will be used by other code. **[4 marks]**

Task 3

Download `cwk1.zip` from the 'Coursework 1' folder in Minerva. Unzip `cwk1.zip` and then delete it. This should leave you with a directory named `cwk1`, containing a Java project managed by the Gradle build system. The code of this project consists of a small Java class named `ArrayChecker`.

[1] Swing is no longer actively developed and has been largely superceded by JavaFX.

1. Identify the 'code smell' that is evident in `ArrayChecker`. Describe briefly how you would refactor the class to remove the smell. **[3 marks]**
2. Study the code for the `ArrayChecker` class and its tests. Identify two other improvements that should be made to this code. **[2 marks]**
3. Make the improvements identified in your answers to the two previous questions. Remember to run the tests at all stages of the refactoring process. You can do this via the Gradle wrapper, with

   ```
   ./gradlew test
   ```

   You can omit the `./` prefix if running on Windows. The tests should pass at all stages.

   When you've finished making all of your changes, enter

   ```
   ./gradlew submission
   ```

   This will create a Zip archive named `cwk1_submission.zip`, containing all of the files that need to be submitted. You will need to submit this Zip archive using Gradescope. **[9 marks]**

Submission

Question answers and code should be submitted to the 'Coursework 1' assignment on Gradescope. You can access this via the link in the 'Submit My Work' folder in Minerva.

The deadline for submission is **10.00 on 3 November 2022**.