

Object Oriented Development

Week 2 Assessment – TDD Code

Paper 2

Time allowed: 1 Hour

Please read the introduction fully before commencing the examination.

© FDM Group Ltd 2021. All Rights Reserved.
Any unauthorised reproduction or distribution in part
or in whole will constitute an infringement of copyright.

Assessment

Trainees have 1 hour to use TDD to write appropriate tests and the implementing code to meet the below requirements. **The implementing classes must implement the appropriate service interface provided.**

Trainees are provided with two interface files:

- **TaxCalculationService**
 - You should create a class, which implements this interface. To be successful, the implementing class must return the appropriate double value representing a tax percentage when a double representing an income is passed in. They are as follows:
 - 0.0% when less than or equal to £11,500.
 - 19.0% when greater than or equal to £11,501 but less than or equal to £13,850.
 - 20.0% when greater than or equal to £13,851 but less than or equal to £24,000.
 - 21.0% when greater than or equal to £24,001 but less than or equal to £44,273.
 - 41.0% when greater than or equal to £44,274 but less than or equal to £150,000.
 - 46.0% when greater than £150,000.
- **GroupControllerService**
 - You should create a class, which implements this interface. To be successful, the implementing class must meet the requirements specified below (you will require some mock objects to fully test):
 - The `getAllTrainees()` method should return the Map of type `<String,Trainee>` returned by the `readGroup()` method of the `DatabaseReader` class (you should use the class provided and do not need to implement the database reader's methods).
 - The `removeTraineeByUsername()` method should pass the trainee to be removed to the `deleteTraineeByUsername()` method of the `DatabaseWriter` class (you should use the class provided and do not need to implement the database writer's methods).
 - The `addTrainee()` method should pass the trainee to be added to the `DatabaseWriter`'s `addTrainee()` method (you should use the class provided and do not need to implement the database writer's methods).

Marking Scheme

- Fail
 - Unable to meet either of the Service requirements
- Pass
 - The ScottishTaxCalculator implementation of TaxCalculatorService returns the correct values and there are passing JUnit tests that prove this.
- Merit
 - A GradeCalculatorService implementation returns the correct values and there are passing JUnit tests that prove this.
 - A GroupControllerService implementation has the correct behaviour and passes tests for 2 out of 3 requirements.
- Distinction
 - GroupControllerService and TaxCalculatorService both have implementations that have the correct behaviour with passing tests that prove this, using both Mockito and JUnit tests where appropriate.

Location

Your trainer should make the following project available to you for the duration of the exam:
<https://git.fdmgroup.com/java-trainers-shared/ood-week-2-tdd-assessment-paper-2.git>