Started on	Saturday, 21 October 2023, 10:00 AM
State	Finished
Completed on	Saturday, 21 October 2023, 12:00 PM
Time taken	2 hours
Grade	**72.00** out of 100.00

Information

INSTRUCTIONS FOR THE EXAMINATION

READ CAREFULLY BEFORE YOU START

- This examination counts for 20% of your final grade.
- The examination is active from 10:00am till 12:30pm.
- The examination stops 120 minutes after you start, or at 12:30, whichever comes first.
- You have only one attempt at this examination.
- All answers are saved automatically as you go through the examination.
- If you need help during the examination, you may go to the posted zoom link.

Question **1**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers)

- [ ] In the Observer pattern, the Subject implements the methods to attach/detach an Observer to the Subject.
- [ ] In the Observer pattern, Observable is an alternate term name for the Subject.
- [x] The Observer pattern uses a push model for the implementation of its notification mechanism.
- [x] In the MVC architecture, the Controller is responsible to update the information on the View.

Mark 0.50 out of 1.00

Question **2**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers)

- [ ] In extreme programming, every continuous integration step produces a deliverable build.
- [ ] In order to be implemented correctly, refactoring requires unit tests for the refactored code.
- [x] In extreme programming, unit tests are written only by a programmer identified as the software tester.
- [ ] In extreme programming, the build plan determines the goals of each successive small release.

Mark 0.50 out of 1.00

Question **3**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers)

- [x] In the Observer pattern, the Subject and Observer classes implement the notification mechanism.
- [ ] In the MVC architecture, a Model object can be connected to more than one View.
- [ ] In the Observer pattern, the notification mechanism calls the update method of every connected observer when the `notifyObservers` method is called upon a state change in the subject class.
- [x] When using the MVC architecture, every Model class must be associated with at least one View class.

Mark 0.50 out of 1.00

Question **4**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] When using incremental development, unit tests should be written only after the build has been approved by the client.
- [x] Unit tests can be written before the tested code is written.
- [ ] All unit test cases should be written after a build is complete, before it is presented to the client.
- [x] A JUnit test that does not include at least one JUnit assert call will automatically fail when executed.

Mark 0.50 out of 1.00

Question **5**

Complete

Mark 2.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] Javadoc should only be written at the end of the production of a build.
- [x] The use of Javadoc increases the understandability of Java code.
- [ ] Any Java block comment is considered as a documentation source by the Javadoc compiler.
- [ ] All Javadoc code should be placed before the `class` keyword of a class declaration.

Mark 1.00 out of 1.00

Question **6**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [x] When using a software versioning repository, complex merge conflicts are best avoided by committing code frequently.
- [ ] In incremental development, each build is managed on a separate branch when using a versioning repository.
- [ ] All merge conflicts can be resolved automatically by a versioning repository during a commit.
- [ ] When using a software versioning repository, each commit corresponds to an extreme programming continuous integration step.

Mark 0.50 out of 1.00

Question **7**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers.)

- [x] Predictive software development methods such as the Rational Unified Process are best suited for very large teams.
- [ ] In agile software development, the only artifacts being produced are directly related to the efficient and sustainable production of implementation code.
- [ ] With adaptive software development models, the team can accurately predict all activities to be performed from the start until the end of the project.
- [ ] Agile software development methods are used mostly by large software development teams.

Mark 0.50 out of 1.00

Question **8**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] In extreme programming, most of the feedback is received at the end of a build when the tests are executed.
- [ ] In extreme programming, a small release is a version of the software that can be used by the client.
- [ ] Refactoring can be used to fix the system's erroneous externally visible behavior.
- [x] In extreme programming, unit tests must be developed and used throughout the production of every build.

Mark 0.50 out of 1.00

Question **9**

Complete

Mark 0.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] The Javadoc documentation is generated as the Java program is compiled for execution.
- [x] The goal in using Javadoc is to decrease the time required to understand the project's code.
- [ ] The Javadoc tool is implemented as a compiler that reads the project’s code and comments.
- [x] Javadoc gets documentation information only from specially-formatted comments in the code base.

Mark 0.00 out of 1.00

Question **10**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers.)

- [x] Adaptive software development models enable a team to start a project with a detailed plan of a software development project from start to finish.
- [ ] The Rational Unified Process is an iterative software development model.
- [ ] In the incremental software development model, a refactoring phase should be done after every build is delivered.
- [ ] In incremental development, a build plan defines the structural design of the build to be implemented.

Mark 0.50 out of 1.00

Question **11**

Complete

Mark 2.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] A unit testing framework automatically runs the test cases when code is committed to a code repository.
- [x] Unit testing is not meant to be used to test system-level use-case scenarios.
- [ ] In JUnit, a test class method annotated with @Before can only refer to static members of its class.
- [ ] Unit testing can be used to ensure that a program is entirely free of bugs.

Mark 1.00 out of 1.00

Question **12**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] A fault is caused by a failure in the code base.
- [ ] JUnit test cases never need to be changed when the tested code is changed.
- [x] In a unit testing framework, a tested unit is a method in the implementation code base.
- [x] Debugging is the process of discovering what caused a failure.

Mark 0.50 out of 1.00

Question **13**

Complete

Mark 0.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers.)

- [ ] Prescriptive models tend to be adaptive models.
- [ ] Prescriptive models focus on being able to plan the future in detail.
- [ ] Predictive models tend to accumulate a large amount of information in a wide variety of artifacts.
- [x] In the waterfall software development model, each phase results in delivering a working version of the system.

Mark -0.50 out of 1.00

Question **14**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers.)

- [ ] An adaptive team can deal with change more efficiently than a predictive team.
- [x] Agile software development methods rely mainly on documentation for communication.
- [x] When using the waterfall model to develop software, it is easy to add or change some of the requirements after the development has started.
- [x] An adaptive team knows very accurately what tasks are to be done in the next few weeks.

Mark 0.50 out of 1.00

Question **15**

Complete

Mark 2.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] When using a revision control system, every programmer is always working on the same version of the developed software.
- [ ] Using a revision control system reduces productivity but makes sure code is not lost.
- [ ] When using a revision control system, every programmer's local copy is automatically kept updated with the commits made by other programmers on the server.
- [x] Revision control systems can be setup to compile and test the code every time a commit is pushed.

Mark 1.00 out of 1.00

Question **16**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [x] Extreme programming is a descriptive software development model.
- [x] Prescriptive models generally define a set of precepts to be followed, without an exact definition of a process.
- [ ] Agile software development does not require planning.
- [ ] An agile team can report exactly what tasks are planned for the entire length of the development process.

Mark 0.50 out of 1.00

Question **17**

Complete

Mark 1.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [x] Extreme programming defines a specific set of coding conventions.
- [ ] Using coding conventions decreases programming productivity but makes the code more readable for the customer.
- [x] Some coding conventions are dependent on the programming language used.
- [ ] All coding conventions can be automatically checked and applied by a compiler.

Mark 0.50 out of 1.00

Question **18**

Complete

Mark 0.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [x] Extreme programming is proposing an incremental model of software development.
- [ ] In extreme programming, small releases is about committing code several times a day.
- [ ] In extreme programming, a build plan determines the goals of each successive small release.
- [x] Extreme programming defines a software development process in terms of roles, activities and artifacts.

Mark 0.00 out of 1.00

Question **19**

Complete

Mark 2.00 out of 2.00

Which of the following affirmations are **FALSE**? (Negative marks for wrong answers)

- [x] Unit testing can be applied to system-level testing activities such as use case scenarios.
- [ ] Testing can reveal only the presence of faults, never their absence.
- [ ] The goal of unit testing is to isolate important parts (i.e. units) of the program and show that the individual parts are free of certain faults.
- [ ] A unit test is a piece of code written by a developer that exercises a very small, specific area of functionality applied to one of the units of the code being tested.

Mark 1.00 out of 1.00

Question **20**

Complete

Mark 2.00 out of 2.00

Which of the following affirmations are **TRUE**? (Negative marks for wrong answers)

- [ ] Coding conventions are only useful when working as a team.
- [ ] Having more comments in our code is always better.
- [x] Using Javadoc is a way to have better commenting conventions.
- [x] Better and consistent code layout leads to decreased debugging time.

Mark 1.00 out of 1.00

Information

INSTRUCTIONS FOR PROGRAMMING QUESTIONS

READ CAREFULLY BEFORE YOU START

- There are some links in the text. Click on them to get access to the code that is mentioned in the text.
- For each question, you have to create and upload a zip file containing the Java code files that answer this question.
- The list of files that you must submit is provided for each question.
- You must have all the specified files and only these files, and with the files' names as specified.
- DO NOT zip all your IDE project folder, as it will be too big to upload. Include only `.java` files.

Question **21**

Complete

Mark 25.00 out of 25.00

The following file contains a Java program:

Click here to download

The GunSlinger class represents a character in a Western movie. Depending on what commands are given to the character, its state will change. As commands are given to the character, the MovieScriptWriter class can be used to print out to the console the script of what is happening in the movie, depending on the state of the characters.

You need to transform this program so that it now uses the MVC architecture and the Observer pattern. From the user's perspective, the resulting program should behave in exactly the same way as the original program. Your solution should consist of the following files:

1. **[5.0 marks] DuelDemo.java** - Main class of the Java program. Creates the Model, View, and Controller objects and connects them together. Starts the execution of the program by calling the controller.
2. **[5.0 marks] GunSlinger.java** - Model class of the program. Implements the logic of providing commands and maintaining the state of the GunSlinger character. This class should not output to the console, and should not call the commands of the Gunslinger character.
3. **[5.0 marks] MovieScriptWriter.java** - View class of the program. Outputs to the console a description of the actions of the Gunslinger character, depending on the state of the character. The displaying should only be triggered by the use of the Observer pattern.
4. **[5.0 marks] ScriptController.java** - Controller class of the program. This is where the commands are sent to the Gunslinger character. It should implement exactly the same sequence of character actions that will lead the program to output the same console output as in the original program.
5. **[2.5 marks] Observer.java** - Implementation of the Observer class exactly as prescribed by the Observer design pattern.
6. **[2.5 marks] Observable.java** - Implementation of the Subject class exactly as prescribed by the Observer design pattern.

You must submit the following files:

1. **DuelDemo.java**
2. **GunSlinger.java**
3. **MovieScriptWriter.java**
4. **ScriptController.java**
5. **Observer.java**
6. **Observable.java**

These files must be put in a zip file named `MVCSolution.zip` and uploaded by drag-and-dropping the zip file in the drop zone below.

MVCSolution.zip

Comment:

Question **22**

Complete

Mark 0.00 out of 10.00

Check the **two** of the following affirmations that are **TRUE**.
You get negative marks if you check those that are false.

- [x] Javadoc gets documentation information only from specially-formatted comments in the code base.
- [x] The use of Javadoc increases the readability of Java code.
- [ ] Any Java block comment is considered as a documentation source by the Javadoc compiler.
- [ ] The Javadoc tool is implemented as a compiler that reads the project's code and comments.

Mark 0.00 out of 1.00

Question **23**

Complete

Mark 25.00 out of 25.00

The following file contains a Java program:

Click here to download

This program implements a simple arithmetic calculator much like a hand-held calculator, where the user enters a number, then an operation, then the second operand of the operation, in which case the calculator display the result of the operation, then the calculator asks for another operation, etc.

You have to write JUnit tests that test the above-mentioned code. The test classes and test methods are specified below. Your solution should consist of the following test classes and test methods:

1. **[9.0 marks]** **CalculatorTestArith.java** - JUnit test class that tests that the arithmetic operations are implementing the right calculations. This should be implemented as four separate test methods, one for each of the addition, subtraction, multiplication and division operator. These four test methods should share the same context, which should be reset before each test method is executed.
2. **[9.0 marks]** **CalculatorTestExpressions.java** - JUnit test class that tests that sequences of operations are implementing the right calculations. This should be implemented as two separate test methods. These two test methods should share the same context, which should be reset before each test method is executed. The first test method should test that the expression 3*4/6-2+2 evaluates to the value 2. The second test method should test the expression 4/2*3+2 evaluates to the value 8.
3. **[7.0 marks]** **CalculatorTestSuite.java** - JUnit test suite that enables to run all the test methods in the above-mentioned test classes.

You must submit the following files:

1. **Calculator.java** (the original tested class file)
2. **CalculatorTestArith.java**
3. **CalculatorTestExpressions.java**
4. **CalculatorTestSuite.java**

These files must be put in a zip file named `JUnitSolution.zip` and uploaded by drag-and-dropping the zip file in the drop zone below.

JUnitSolution.zip

Comment:

◄ Quiz Sample

Jump to...

Design Patterns ►