Calitate și testare software, Tip-C, Sem-2, Zi (2023-2024)

Started on	Tuesday, 4 June 2024, 6:38 PM
State	Finished
Completed on	Tuesday, 4 June 2024, 7:30 PM
Time taken	52 mins 12 secs
Marks	37.10/50.00
Grade	**6.68** out of 9.00 (**74.2**%)

Quiz navigation

1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50

Show one page at a time
Finish review

Question **1**
Complete
Mark 1.00 out of 1.00
Flag question

Which principle of Clean Code is violated in the Angajat (Employee) class in the next image

```
public abstract class Angajat {
    private String nume;
    private float salariu;
    private String functie;

    public abstract float calculeazaSalariu() ;
    public abstract List<Angajat> getAngajatiFromFile();
    public abstract saveAngajatiInBD(List<Angajat> angajati);
    public abstract void avansareInFunctie();
}
```

a. DRY - Don't Repeat Yourself
b. Single Responsability
c. Open-Closed
d. Liskov Substitution
e. Dependency Inversion

Question **2**
Complete
Mark 0.00 out of 1.00
Flag question

Considering the fact that the power method raises a number to a certain power, the following unit test type is (choose one from the next CORRECT or Right-BICEP options):

```
@Test
public void testOne() {
    double number = 3;
    double exponent = 2;
    double value = Main.power( number: 3, exponent: 2);
    double result = Math.sqrt(value);
    Assert.assertEquals( expected: 3, result,  delta: 0.0001);
}
```

a. Right
b. Inverse Relationship
c. Cross-check
d. Correct
e. Range

Question **3**
Complete
Mark 1.00 out of 1.00
Flag question

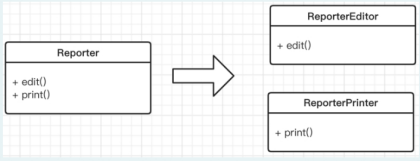
In the following figure, the implementation of a Singleton Registry was needed.
Identify issues within this Singleton Registry implementation:

```
public class Produse implements ProdusGeneric {

    private String denumire;
    private float stoc;

    private static Map<String, Produse> produseUnice = new HashMap<>();

    private Produse(String denumire, float stoc) {
        this.denumire = denumire;
        this.stoc = stoc;
    }

    public static Map<String, Produse> getInstanta(String denumire) {
        if(produseUnice.containsKey(denumire)) {
            produseUnice.put(denumire,produs);
        }
        return produseUnice;
    }
```

a. Colectia din cadrul clasei trebuie sa fie o lista simpla sau un vector, insa in cadrul implementarii este un Map (HashMap)
b. The getInstanta() method must be final, but in the implementation it is static
c. The static function getInstanta() must return a single instance, but in the implementation it returns a collection
d. The static function getInstanta() must add the new product to the collection only if it does not exist, but the implementation adds it if it exists.
e. The constructor of the class must be public, but in the implementation it is private

Question **4**
Complete
Mark 1.00 out of 1.00
Flag question

What SOLID principle is used to change the following solution from the variant on the left to the one on the right?



a. Dependency Inversion Principle (DIP)
b. Liskov Substitution Principle (LSP)
c. Open-Closed Principle (OCP)
d. Interface Segregation Principle (ISP)
e. Single Responsibility Principle (SRP)

Question **5**
Complete
Mark 0.00 out of 1.00
Flag question

What is the "clean code" principle that classes must depend on abstractions, not concrete implementations?

a. Liskov Substitution Principle (LSP)
b. Dependency Inversion Principle (DIP)
c. Interface Segregation Principle (ISP)
d. Open-Closed Principle (OCP)

Question 6
Complete
Mark 1.00 out of 1.00
Flag question

For the method public float method(int a, int b), having as limitations of the accepted values for variable b any integer value included in the interval (10,30], which of the following method calls generates boundary type testing?

- [x] a. 30
- [ ] b. 29
- [ ] c. Integer.MAX_VALUE
- [ ] d. 10
- [x] e. 11

Question 7
Complete
Mark 0.50 out of 1.00
Flag question

The following scenario is considered:

A person can apply for the change of driver's license only at the special dedicated counter. From here the request will be analyzed by the duty officer who will ask the printer to print a new permit if this is legal. From the printing office, the permit will arrive at the pick-up counter, where it can be picked up by the applicant. The method of evaluating the possibility of issuing a new permit, as well as all the stages it goes through from the submission of the application to the issuance, are not transparent for the applicant.

What combination of Design Patterns can you use to implement this scenario?

- [ ] a. Composite
- [ ] b. Proxy
- [ ] c. Chain of Responsibility
- [ ] d. Command
- [ ] e. Strategy
- [ ] f. Prototype
- [ ] g. Adapter
- [ ] h. Observer
- [ ] i. State
- [x] j. Facade

Question 8
Complete
Mark 0.00 out of 1.00
Flag question

How many unit tests will be marked as successfully (green check icon) considering the next test case?

```
public class TestCase {

    @Test
    public void test1() {
        fail();
    }

    @Test
    public void test2() {
    }

    @Test
    public void test3() {
        Object o1 = new Object();
        Object o2 = o1;
        assertSame(o1, o2);
    }
}
```

- [ ] a. running the tests will result in a runtime error
- [x] b. 1
- [ ] c. 0
- [ ] d. 3
- [ ] e. 2

Question 9
Complete
Mark 1.00 out of 1.00
Flag question

Within an e-commerce web application, the user has the option to select the viewing layout for the products within the store. Thus, he can choose between: displaying products in ascending price order, descending price order, displaying only products that have at least one review. What is the design pattern that best models the stated scenario?

- [ ] a. Command
- [ ] b. Template
- [ ] c. Observer
- [x] d. Strategy
- [ ] e. State

Question 10
Complete
Mark 1.00 out of 1.00
Flag question

The 70/20/0 rule refers to

20	percentage of total development time
0	the number of failed unit tests
70	percent code coverage

Question 11
Complete
Mark 0.67 out of 1.00
Flag question

Which of the following patterns (select one or more options) implements the callback principle, decoupling the module you manage from the actual implementation of the action?

- [ ] a. Builder
- [ ] b. Observer
- [ ] c. Adapter
- [ ] d. Prototype
- [ ] e. Memento
- [x] f. Strategy
- [ ] g. Flyweight
- [ ] h. Proxy
- [ ] i. Decorator
- [x] j. State
- [ ] k. Composite

Question 12
Complete
Mark 1.00 out of 1.00
Flag question

What "clean code" principle does the following implementation violate:

```
public class Customer {
    private String name;
    private String address;

    public void saveToDatabase() {
        // Cod pentru salvarea informatiilor clientului in baza de date
    }
    public void sendEmail() {
        // Cod pentru a trimite un email
    }
}
```

- [ ] a. Dependency Inversion Principle (DIP)
- [x] b. Single Responsibility Principle (SRP)
- [ ] c. Interface Segregation Principle (ISP)
- [ ] d. Open-Closed Principle (OCP)

Question 13
Complete
Mark 1.00 out of 1.00
Flag question

Associate each Design Pattern from the list below with the problem it solves:

Flyweight	Objects share common areas and the total memory space occupied can be reduced
Factory	Creation of objects that belong to the same family of objects
Decorator	The object must receive new methods or attributes at runtime
Prototype	Creating objects is expensive in terms of time or total memory used
Composite	Modeling a hierarchical structure

Question 14

Which of the following statements about the Cross-Check Testing Principle from Right-BICEP are correct

Complete
Mark 0.67 out of 1.00
Flag question

a. It allows the use of parameterized tests to check the consistency of the results.
b. Performs validation of test results by comparing them with results obtained by an external reference system.
c. Validates the results of a test with predefined values.
d. Cross-Check testing involves checking the results of a method using other independent methods.
e. Reduce necesitatea de a scrie documentație detaliată pentru teste.
f. It involves ensuring that a test covers all branches of the code under test
g. Cross-Check testing allows different algorithms to be used to calculate the same result.

Question 15
Complete
Mark 1.00 out of 1.00
Flag question

To which design pattern should the following method be assigned, taking into account the main problem that each design pattern addresses?

```
public void transmiteComenzile() {
    for(IComanda c: this.listaComenzi)
        c.prelucreaza();
    this.listaComenzi.clear();
}
```

a. Memento
b. Composite
c. Command
d. Chain of resposability
e. Observer

Question 16
Complete
Mark 1.00 out of 1.00
Flag question

For the method public boolean method(List<Integer> listGrade) which returns true if the list contains at least 2 passing grades, what is the minimum number of tests of the Cardinality type that must be performed to cover all possible cases?

a. 1
b. 2
c. 4
d. 3
e. 5

Question 17
Complete
Mark 1.00 out of 1.00
Flag question

Choose the annotations used in JUnit 4 for the following elements:

Creating a test	@Test
Removing certain categories from a custom suite	@ExcludeCategory
Creating a setup	@Before
Adding TestCases in a suite	@SuiteClasses
Adding certain categories in a custom suite	@IncludeCategory

Question 18
Complete
Mark 1.00 out of 1.00
Flag question

The Singleton design pattern is used when:

a. an object is created based on another object
b. an object from a class framework is created
c. an intermediate object is created compared to the original object
d. a unique object of a class is created
e. a verification object of the original object is created
f. a complex object is created
g. a single instance of a class is needed

Question 19
Complete
Mark 0.00 out of 1.00
Flag question

You are doing a code review for a colleague. What are the Clean Code rules/principles that must be implemented in the code and from the perspective of which you can consider that the solution is clean. Select one or more answers.

a. The code must contain comments that explain how the function is implemented
b. Functions must return null as a result, if it does not exist or cannot be determined
c. The functions must validate the value of the input parameters
d. Functions must NOT return null as a result if it does not exist
e. The source code of a function must fit on a screen (minimum 14 inches)
f. Classes must contain accessor methods at the expense of public attributes
g. Functions must do one thing
h. Functions must throw exceptions if they cannot process the data
i. Use generic names for variables and functions
j. It uses names of variables and functions that are related to the domain of the solved problem
k. Use variable and function names that reveal intent
l. Functions should be small (as number of lines of source code)
m. Classes must be small (in number of lines of source code)
n. The functions must return error codes if they cannot process the data

Question 20
Complete
Mark 1.00 out of 1.00
Flag question

In the restaurant, customers have the opportunity to choose one of the products available in its menu. However, over time, it has been noticed that there are additional requests for a selection of products with Spanish and Italian specifics. Thus, for each product available in the restaurant's menu, when it is prepared, an additional topping related to the selected specifics can be added. The selection of the culinary specialty does not lead to a change in the basic price of the dish. Customers can also order basic items from the menu, without having to make any changes to the basic recipe. What is the design pattern that shapes the existing problem from the statement?

a. Flyweight
b. State
c. Factory
d. Decorator
e. Composite

Question 21
Complete
Mark 0.67 out of 1.00
Flag question

Which of the following Design patterns are also considered Wrappers for the original classes?

a. Template
b. Singleton
c. Prototype
d. Adapter
e. Observer
f. Abstract Factory
g. Proxy
h. Facade

Question 22
Complete
Mark 1.00 out of 1.00
Flag question

Using JUnit 4, if you need to connect to a database to run several tests, in which standard JUnit4 method (marked with one of the following annotations) would you make the connection so that it is optimal?

a. @DBConnect
b. @BeforeClass
c. @TearDownAfterClass
d. @Test
e. @SetUpBeforeClass

Question 23
Complete
Mark 1.00 out of 1.00
Flag question

Using the Proxy design pattern is recommended when:

a. It is desired to add new attributes to an existing class
b. It is desired to create a unique object that will perform the action
c. It is desired to adapt some existing functionalities to new contexts
d. It is desired to restrict/control the call of an existing functionality

e. It is desired to simplify the calling of an existing functionality

Question 24
Complete
Mark 1.00 out of 1.00
Flag question

Which of the following statements about the Adapter design pattern are true:

- [x] a. It can be implemented using composition (HAS-A) or inheritance (IS-A).
- [x] b. It allows using a wrapper class to convert an old interface to a new one.
- [ ] c. Allows extending a class to add new functionality.
- [ ] d. It allows adding new functionality to an existing class without modifying it.
- [x] e. Allows incompatible interfaces to be used together as one.
- [ ] f. Allows objects to change their behavior at runtime.

Question 25
Complete
Mark 0.50 out of 1.00
Flag question

Develop a software solution for a car wash.
Within it, the process of washing the car involves going through some steps in a well-established order: adding degreaser, washing, adding foam, rinsing, wiping. This process is similar for all laundromats.
The solution must be modified so that it only starts if there is enough degreaser and enough detergent to foam. Otherwise, the car wash process will not start, waiting for the tanks to be filled.
What are the 2 design patterns that optimally solve these 2 problems?

- [ ] a. Composite
- [ ] b. Prototype
- [ ] c. State
- [ ] d. Builder
- [ ] e. Proxy
- [ ] f. Command
- [ ] g. Singleton
- [ ] h. Flyweight
- [ ] i. Chain of Responsability
- [ ] j. Decorator
- [ ] k. Observer
- [x] l. Template
- [ ] m. Factory
- [ ] n. Strategy
- [ ] o. Adapter
- [ ] p. Memento

Question 26
Complete
Mark 1.00 out of 1.00
Flag question

Select from the code examples below only the variant(s) that correctly implement the "singleton" design pattern:

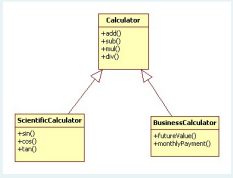
a.
```
public class Server {
    private static Server instance = null;
    private Server(){}
    public static synchronized Server getInstance()
    {
        if(Server.instance == null)
        {
            Server.instance = new Server();
        }
        return Server.instance;
    }
    public void run() { System.out.println("Server is running"); }
}
```

b.
```
public class Server {
    private static Server instance = null;
    private Server(){}
    public static synchronized Server getInstance()
    {
        if(Server.instance == null)
        {
            Server.instance = new Server();
        }
        return new Server();
    }
    public void run() { System.out.println("Server is running"); }
}
```

c.
```
public class Server {
    private static Server instance = null;
    public Server(){}
    public static synchronized Server getInstance()
    {
        if(Server.instance == null)
        {
            Server.instance = new Server();
        }
        return Server.instance;
    }
    public void run() { System.out.println("Server is running"); }
}
```

Question 27
Complete
Mark 1.00 out of 1.00
Flag question

Which SOLID principle is used by the following solution?



a. Open-Closed Principle (OCP)
b. Interface Segregation Principle (ISP)
c. Liskov Substitution Principle (LSP)
d. Dependency Inversion Principle (DIP)
e. Single Responsibility Principle (SRP)

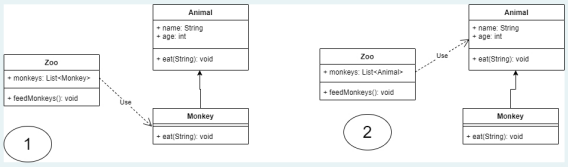
Question 28
Complete
Mark 0.60 out of 1.00
Flag question

Associate the situations described below with the related Right-BICEP principles.

E — Using expected from JUnit4
B — Values of -1, 0 and 1 for a person's age
C — Using the radical of degree 2 to test a squaring function
P — Using timeout from Junit4
I — Using a quick sort to test a bubble sort algorithm

Question 29
Complete
Mark 1.00 out of 1.00
Flag question

According to which principle of Clean Code diagram 1 must be transformed into diagram 2?



a. Single responsibility
b. Interface segregation
c. Dependency inversion
d. Open-closed
e. Liskov substitution

Question 30
Complete
Mark 1.00 out of 1.00
Flag question

What is the Singleton design pattern category implemented in the following class?

```
public class Clasa {
private static Clasa instance;

    private Clasa() {

    }

    static {
        try {
            instance=new Clasa();
        } catch(Exception e) {
            throw new RuntimeException("Exceptie");
        }
    }

    public static Clasa getInstance() {
        return instance;
    }
}
```

a. Inner static helper
b. Eager initialization
c. Singleton Collection
d. Singleton Static Block
e. Lazy initialization

Question 31
Complete
Mark 1.00 out of 1.00
Flag question

It is considered a TestCase with the following methods. After running this TestCase, how many tests will be run by a specific JUnit4 Runner?

```
@Test(timeout=10)
public void test1() {
    System.out.println("        Test 1");
}


public void test2() {
    System.out.println("        Test 2");
}

@Test(expected = Exception.class)
public void test3() {
    System.out.println("        Test 3");
}
```

a. 0
b. 3
c. 1
d. 2

Question 32
Complete
Mark 1.00 out of 1.00
Flag question

You develop a Visual Code type application for editing source code. What design patterns should you implement to optimally solve the following requirements (select one or more patterns)?

- the application menu has a tree like structure, being based on options and groups of options; the solution must allow the flexible scaling of the menu and the association of access roles to certain options
- the interface contains many graphic elements (pictograms) that are repeated; for example, the same icon is used for several options in the menu or in the graphical interface; the memory occupied by the application must be minimized
- the user can receive various notifications depending on the operations he performs directly or depending on the services offered by the application; the user can subscribe or unsubscribe from certain notifications in the application settings

a. Decorator
b. Command
c. Builder
d. Adapter
e. Proxy
f. State
g. Strategy
h. Observer
i. Flyweight
j. Composite
k. Prototype

Question 33
Complete
Mark 0.75 out of 1.00
Flag question

Match the variables with the used naming standard:

saveBtn	lowerCamelCase
numberOfBigNumbers	lowerCamelCase
ObtainedTemperature	UpperCamelCase
expected_result	Snake case

Question 34
Complete
Mark 1.00 out of 1.00
Flag question

You make an MVC (Model View Controller) application. What combination of Design Patterns can you use to decouple the View from the Controller when it comes to updating some fields (eg: loading a form with data from the database) or executing some actions (clicking on a button)?

a. Proxy
b. Facade
c. Memento
d. Strategy
e. Adapter
f. Observer
g. Decorator
h. Command

Question 35
Complete
Mark 0.00 out of 1.00

You develop an application that allows electronic payment of invoices. What design patterns should you implement to optimally solve the following requirements (select one or more patterns)?

- payment can be made using any existing processor, such as Visa, Mastercard, etc. The module that executes the payment is implemented independently of the processor. The processor is indicated when the bank card data is entered. The solution must allow the flexible addition of new processors in the future

Flag question

- the application must allow additional operations to be carried out before the payment reaches the processor. The solution must allow the addition of these additional operations (verifications, validations, data analysis, etc.) without affecting the main payment execution flow
- the application can allow the user to schedule payments for a future date so that they are not made instantly.

a. Adapter
b. Builder
c. Composite
d. Strategy
e. State
f. Flyweight
g. Decorator
h. Prototype
i. Proxy
j. Command
k. Observer

Question **36**
Complete
Mark 1.00 out of 1.00
Flag question

You want to test a method that computes the average value of an input collection using the Cardinality perspective (from CORRECT). What tests can you do from this perspective (choose 1 or more)

a. A test with an empty collection
b. A test with a collection that has {10}
c. A test with a null reference
d. A test with a collection that has {10, 20}
e. A test with a collection that has {1, 2, 3, 4, 5,....., 100}

Question **37**
Complete
Mark 0.00 out of 1.00
Flag question

It has been observed that when the internet connection is weak, the loading of the resources of a web page takes a very long time. Thus, the decision was made to optimize this process by creating a filter on the loading elements. A web page is represented by a list of Item type objects (paragraphs, images, etc.). Filtering consists of several stages such as: removing image objects, changing paragraph settings by reducing them to black only, without other settings such as bold, italics, etc. It is recognized that these steps can be interchanged and also that additional filtering can be added if the present implementation does not achieve the desired optimum. What is the design pattern that shapes the existing problem in the statement?

a. Flyweight
b. State
c. Strategy
d. Chain of responsability
e. Proxy

Question **38**
Complete
Mark 0.75 out of 1.00
Flag question

It is considered the Composite class from the implementation of the design pattern with the same name.
Match the terms below according to the correct implementation of this pattern.

```
public class Composite implements [Composite] {
    [List] <Component> components = new [ArrayList] <Component>();

    public void remove(Component component) {
        components.remove(component);
    }

    public void add(Component [component] ) {
        components.add(component);
    }
}
```

[Component] [Map] [Client] [Leaf] [MapArray]

Question **39**
Complete
Mark 0.00 out of 1.00
Flag question

The following method and related tests are given.
Which of the unit tests fails?

```
public int getFrecventaValoareMax(int[] v){
    int ct=1;
    int max=v[0];
    for(int i=0;i<v.length;i++) {
        if(max>=v[i]) {
            max=v[i];
            ct++;
        }
    }
    return ct;
}
```

```
@Test
public void test5() {
    int[] v=new int[] {};
    assertEquals(0, obj.getFrecventaValoareMax(v));
}

@Test
public void test6() {
    int[]v = new int[] {9,10,10,10,15,15};
    assertEquals(2,obj.getFrecventaValoareMax(v));
}

@Test
public void test7() {
    int[] v= new int[] {10,5,12,12};
    assertEquals(2,obj.getFrecventaValoareMax(v));
}
```

a. test5 and test7
b. test7
c. test5 and test6
d. all tests
e. test5

Question **40**
Complete
Mark 1.00 out of 1.00
Flag question

In which of the following situations would it be most appropriate to use the Prototype design pattern?

a. When we want to organize code structure by using abstract classes
b. When we want to modify the behavior of a class at runtime
c. When creating the object is expensive and requires a lot of time and resources and we already have a similar object
d. When we want to implement flexible communication between objects via an event channel

Question **41**
Complete
Mark 1.00 out of 1.00
Flag question

For the next method

```
public float ceaMaiMicaSolutie(float a, float b, float c) throws Exception {
    float delta = b*b-4*a*c;
    if(a!=0) {
        if(delta>=0)
        return (float) ((-b-Math.sqrt(delta))/(2*a));
        else
            throw new Exception("Nu exista solutii reale");
    }
    else
        throw new Exception("Nu este ecuatie de gradul doi");

}
```

What is the minimum number of tests that must be implemented to achieve 100% code coverage?

a. 1
b. 2
c. 3
d. 100% code coverage cannot be achieved

Question 42
Complete
Mark 1.00 out of 1.00
Flag question

Which Design Pattern implementation can the following class be part of?

```
public class History {
    2 usages
    private Stack<String> messages = new Stack<>();

    public void saveLastMessage(String message) {
        messages.push(message);
    }

    public String getLastMessage() {
        return messages.pop();
    }
}
```

a. Mediator
b. Template method
c. State
d. Proxy
e. Bridge
f. Interpreter
g. Memento
h. Prototype

Question 43
Complete
Mark 0.00 out of 1.00
Flag question

Select the differences between the State and Strategy patterns

a. Only State outsource the implementation of different actions
b. For strategy the logic is that the client changes/selects the action to be executed
c. For state the logic is that the object can change the action to be executed
d. Only Strategy outsource the implementation of different actions
e. Strategy is a structural pattern, while State is a behavioral one
f. Strategy is a behavioral pattern, while State is a structural one

Question 44
Complete
Mark 1.00 out of 1.00
Flag question

You are developing a software that will be installed on smart drones. You have these requirements

- Different types of drones or operational modes require different actions to execute a command. The software should allow the user to easily change the operational mode (or add new modes). The solution must not generate any source code changes in the default mode
- You want to add logging to various components of your drone control software. All log messages (from all modules) must be stored in the same log file and software developers should not be able to create additional files.

What design patterns should you use in order to efficiently implement the previous requirements (one design pattern per requirement) ?

a. Strategy
b. Flyweight
c. Composite
d. Command
e. Factory
f. Adapter
g. Chain of Responsibility
h. State
i. Decorator
j. Proxy
k. Singleton

Question 45
Complete
Mark 0.00 out of 1.00
Flag question

Which Design Pattern does the following constructor requires?

```
public Car(CarType carType, int noOfSeats, Engine engine, Transmission transmission, String navigationProducer) {
    this.carType = carType;
    this.noOfSeats = noOfSeats;
    this.engine = engine;
    this.transmission = transmission;
    this.navigationProducer = navigationProducer;
}
```

a. Prototype
b. Simple Factory
c. Factory Method
d. Abstract Factory
e. Builder

Question 46
Complete
Mark 1.00 out of 1.00
Flag question

What will the following test case display when it is executed (assuming that the unit tests will be executed in the order written)?

```
public class TestMetode {

    @BeforeClass
    public void setUpBeforeClass() {
        System.out.print("Before ");
    }

    @Test
    public void test1() {
        System.out.print("Test1 ");
    }

    @Test
    public void test2() {
        System.out.print("Test2 ");
    }

    @After
    public void tearDown() {
        System.out.print("After ");
    }
}
```

a. Before Test1 Before Test2 After
b. Before Test1 After Test2 After
c. Before Test1 After Test2
d. Before Test1 After Before Test2 After
e. Before Test1 Test2 After

Question 47
Complete
Mark 0.00 out of 1.00
Flag question

Select from the list below the SOLID principles that emphasize the importance of designing classes and interfaces that are cohesive and focused on specific tasks.

a. Open-Closed Principle (OCP)
b. Single Responsibility Principle (SRP)
c. Interface Segregation Principle (ISP)
d. Dependency Inversion Principle (DIP)
e. Liskov Substitution Principle (LSP)

Question 48

Determine for each design pattern which category it belongs to:

Complete

Mark 1.00 out of 1.00

Flag question

Singleton	creational
Facade	structural
Chain of Responsability	behavioral
Factory Method	creational
Strategy	behavioral
Adapter	structural

Question **49**

Complete

Mark 1.00 out of 1.00

Flag question

It is desired to implement the software for an intelligent door camera. It is desired that when someone arrives in front of the door, the camera will notify all the owners of the house that someone has come to the door. The solution allows the addition of new people within the owners who are announced and also the removal of certain owners from this collection.

Every time a person passes in front of the door, the viewer has the possibility to realize a photo, to take a 5-second video or to take a 10-second video. The user has the possibility to establish and modify whenever he wants the way in which the viewfinder saves the photos or videos.

The photos or videos taken are saved in the internal memory. If the internal memory is fully occupied, they are saved on an SD memory card. When there is no more space on this SD card either, the photos are saved in the cloud. In the cloud it ends up being saved only if there is not enough space in the internal memory and on the SD card, because the costs are high.

What are the 3 design patterns that optimally solve these 3 problems ?

- [ ] a. Factory
- [x] b. Strategy
- [ ] c. Memento
- [ ] d. Composite
- [ ] e. Builder
- [ ] f. Decorator
- [ ] g. Prototype
- [ ] h. Singleton
- [ ] i. Proxy
- [ ] j. Command
- [ ] k. Flyweight
- [ ] l. Template
- [x] m. Observer
- [ ] n. State
- [x] o. Chain of Responsability
- [ ] p. Adapter

Question **50**

Complete

Mark 1.00 out of 1.00

Flag question

Which annotation should be used to specify that a method should be executed before each test in a test class in JUnit 4?

- ( ) a. @Before
- ( ) b. @BeforeTest
- ( ) c. @Setup
- ( ) d. @BeforeClass
- ( ) e. @PreTest