Software Design 23/24: Exam

Justus Bogner <j.bogner@vu.nl>
Ivano Malavolta <i.malavolta@vu.nl>

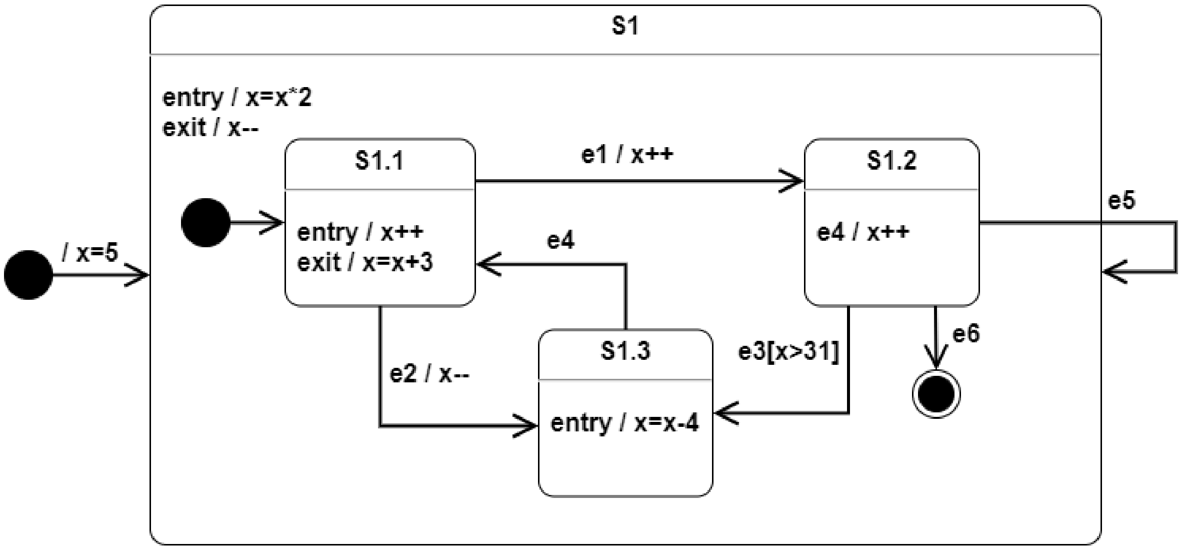
Shared to allow students to prepare for the resit exam.

20 single-choice questions, at least 12 correct answers required for passing (6.0)

Q1 (1 point): Consider the following state machine diagram representing the behavior of a system with a variable x. Assume that the state machine started and then this event sequence occurs:

e2, e4, e1, e5, e1, e3

What is the value of variable x after that?



A. 31
B. 27
C. 30
D. 26

Q2 (1 point): Which design pattern is appropriate for implementing the following scenario? "A social media platform allows users to follow their favorite artists and to receive notifications whenever the artists release new music or announce upcoming concerts. The platform should notify all followers of an artist whenever such an event occurs."

A. Iterator
B. Observer
C. Adapter
D. Decorator

Q3 (1 point): Which design principle is violated in the following Java code snippet?

```
public class FileManager {
    public void saveFile(String fileName, String content) {
        // save file logic
    }

    public void loadFile(String fileName) {
        // load file logic
    }

    public void deleteFile(String fileName) {
        // delete file logic
    }

    public void encryptFile(String fileName, String algorithmName) {
        // encryption logic
    }
}
```

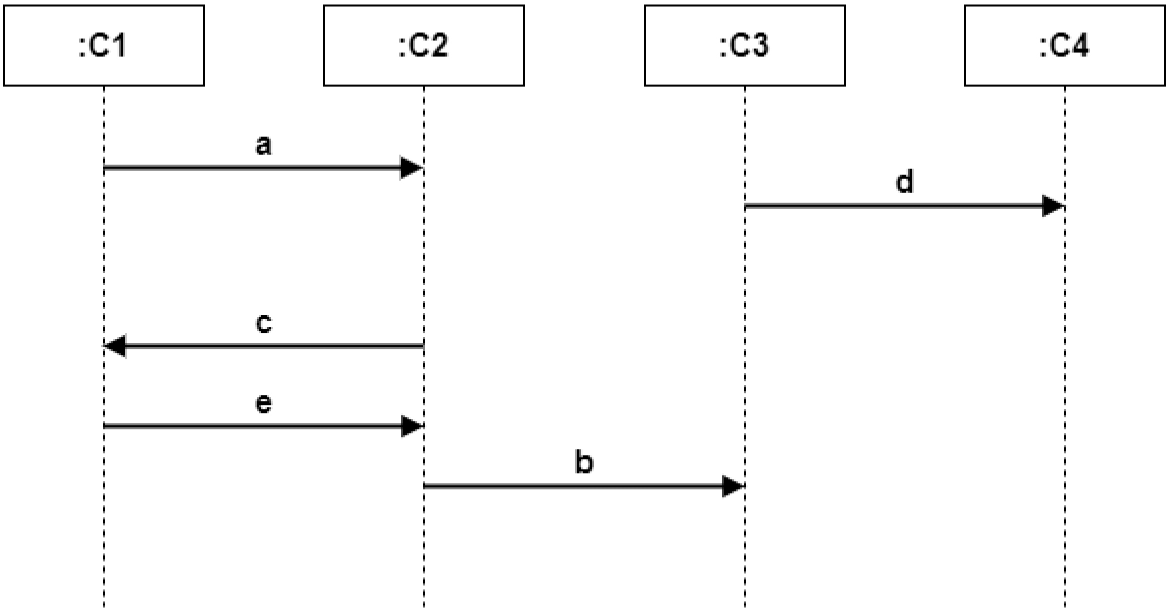
A. Information hiding
B. Law of Demeter
C. Single responsibility principle
D. Interface segregation principle

Q4 (1 point): Which design principle is applied in the below Java code snippet?

```
public class Car {
    private final String carID;
    private final Owner owner;

    public Car(String carID, Owner owner) {
        this.carID = carID;
        this.owner = owner;
    }

    public String getCarID() {
        return carID;
    }

    public Owner getOwner() {
        return owner;
    }
}
```

A. Immutability
B. Encapsulating inherent complexity
C. Liskov substitution principle
D. Inversion of Control

Q5 (1 point): Given the following sequence diagram, which trace is **not** possible?

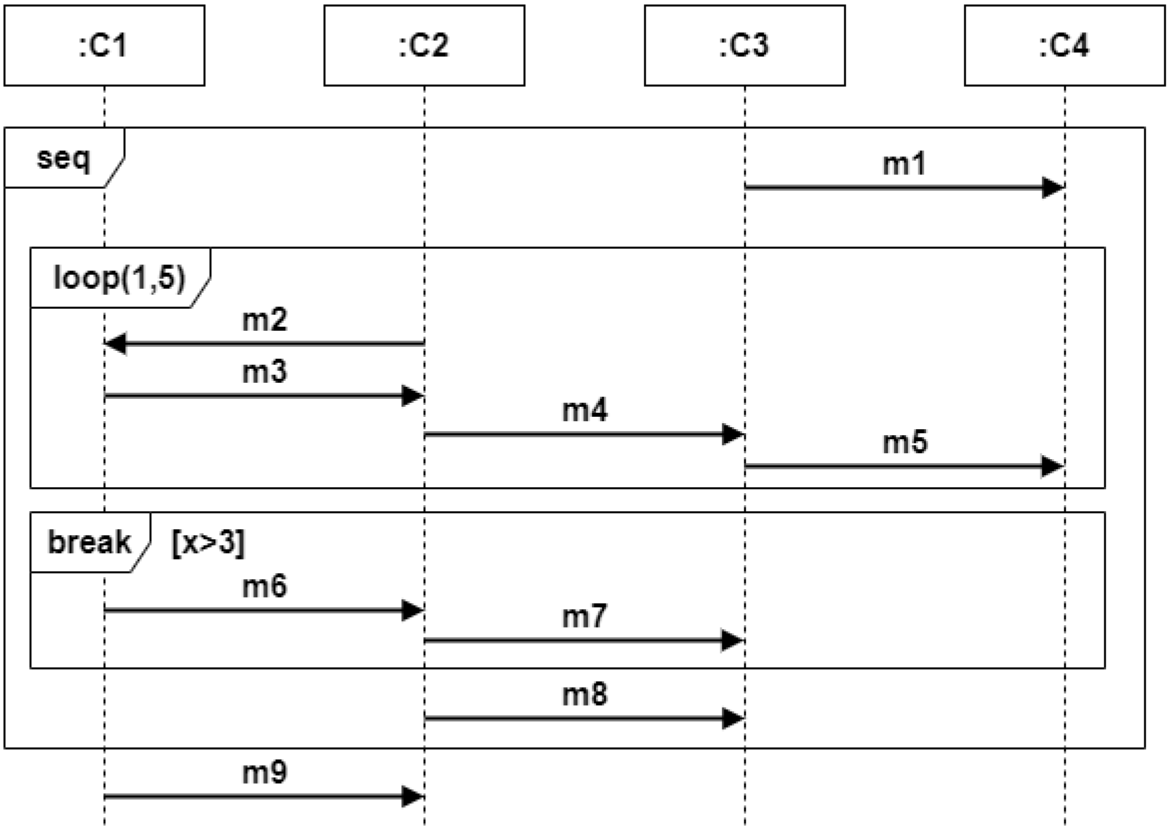


A. a → c → e → d → b
B. d → a → c → e → b
C. a → d → e → b → c
D. a → c → d → e → b

Q6 (1 point): Consider the following relationship description between two classes: “In a company, an employee always has exactly one manager that is responsible for the yearly performance evaluation. It should never happen that an employee has no manager, i.e., an employee cannot exist without a manager.” How would you model this relationship in a class diagram?

A. Composition
B. Shared aggregation
C. Association
D. Inheritance

Q7 (1 point): Given the following sequence diagram, which statement is **false**?



A. Regardless of the value of x, the loop is always executed exactly five times.
B. If the value of x after the loop is greater than 3, m9 is never executed.
C. If x is greater than 3 after the loop, m6 and m7 will be executed.
D. Removing the seq fragment from the diagram would change the execution semantics.

Q8 (1 point): Which of the following statements about generalization in class diagrams is **true**?

A. Every instance of a superclass is also an indirect instance of its subclasses.
B. All attributes of the superclass (regardless of their visibility) are inherited by its subclasses.
C. Instantiating an abstract class is possible, unless it is marked as static.
D. Having deep inheritance hierarchies might impact the maintainability of the system.

Q9 (1 point): Which of the below state machine diagrams best describes the following scenario? "A smart garage door system always starts in the state Open. When the owner's smartphone disconnects from the WiFi (e1) or when a button is pressed in the app (e5), the system changes to the Closing state. In both cases, however, it first checks if no object is blocking the path of the garage door before transitioning (g1). When the door is extended to full length (e2), the garage light is switched off (a1) and the system changes to the state Closed. From there, pressing another button in the app (e3) transitions the system to the Opening state. When the door has been completely retracted (e4), the light is switched on (a2) and the system changes to the state Open again."

<do not randomize order of options>

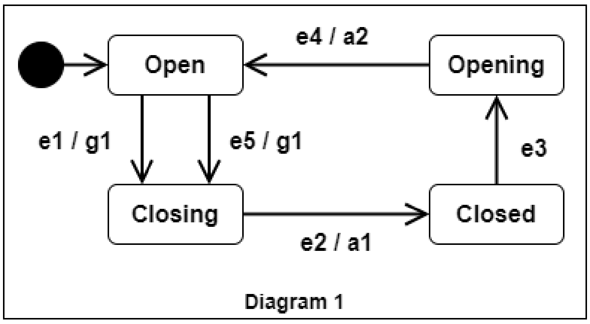


Diagram 1

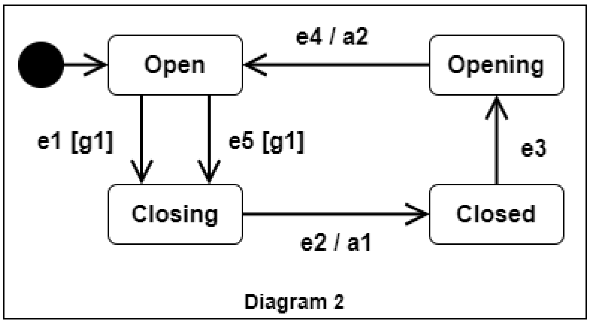


Diagram 2

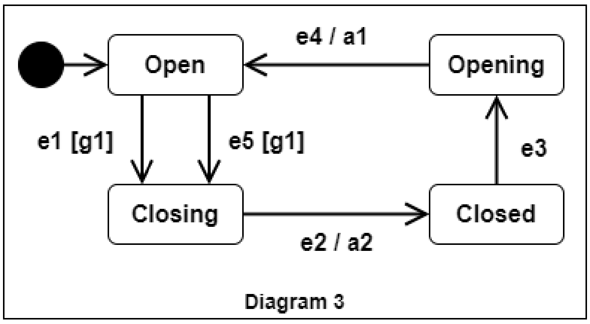


Diagram 3

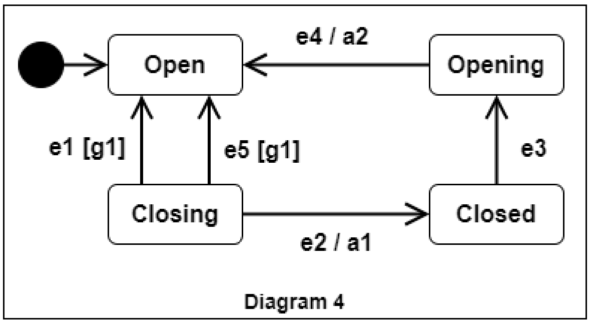


Diagram 4

A. Diagram 1
B. Diagram 2
C. Diagram 3
D. Diagram 4

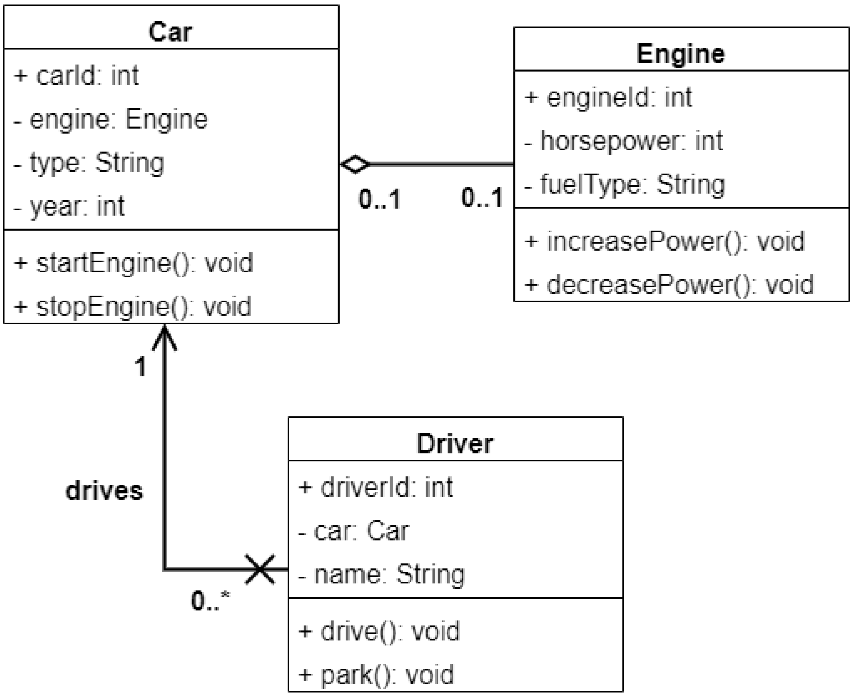
Q10 (1 point): Which design pattern is appropriate for implementing the following scenario? “A complex application has an extensive number of configuration options that are loaded from a config file before startup. Some of these options can be changed at runtime. The loaded configuration needs to be made available to many other classes, but it is also important that only a single version of the configuration exists.”

A. Factory Method
B. Decorator
C. Flyweight
D. Singleton

Q11 (1 point): Which statement about UML package diagrams is **false**?

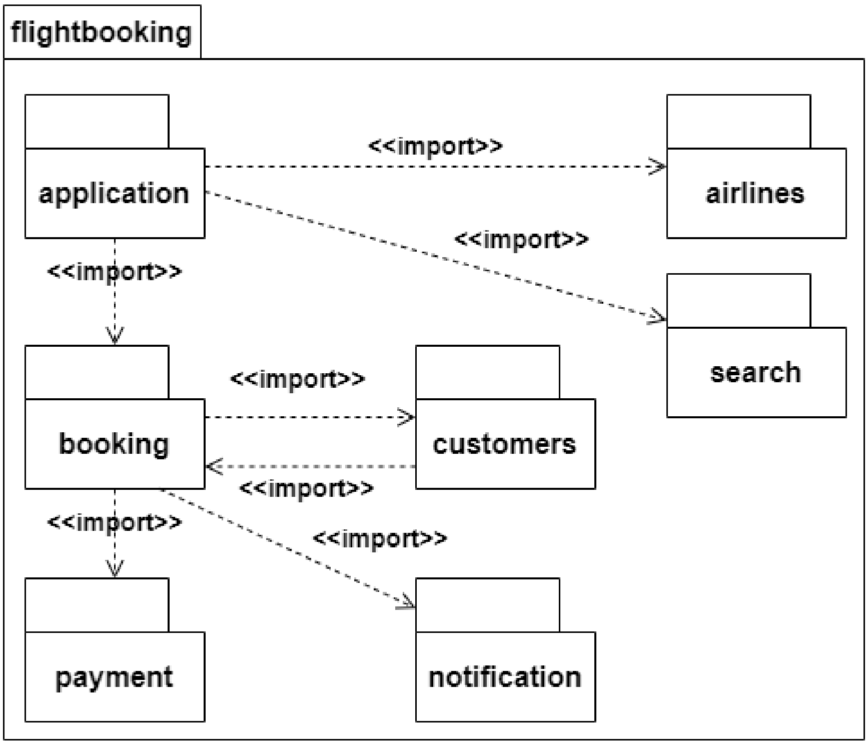
A. A class can be in exactly one package.
B. Packages are meant to improve understanding and navigating through a project.
C. Package by layer is great for isolating change and therefore recommended for large projects.
D. A package groups various models or model elements together.

Q12 (1 point): Which conceptual error exists in the below class diagram?



A. In reality, a Car cannot have more than one Driver, so the multiplicity should be changed from 0..* to 0..1.
B. A Car cannot exist without an Engine, so the relationship should be a composition instead.
C. Each driver has exactly one Car, so the relationship should be a composition instead.
D. The indicated navigability requires that a Car has a reference to a Driver.

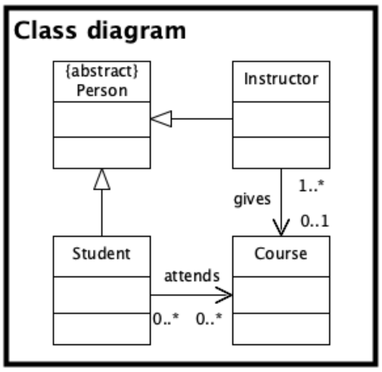
Q13 (1 point): Which statement about the below package diagram is **true**?

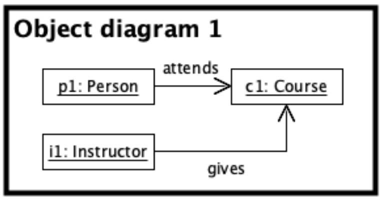


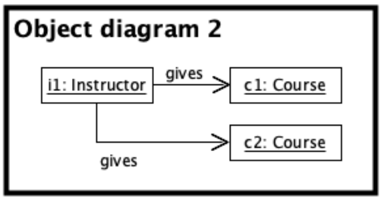
A. The hierarchical imports indicate that the system is packaged by layer.
B. The cyclic dependency between `booking` and `customers` is not problematic because the `customers` package does not import any other packages.
C. The application package imports three classes.
D. The naming after domain concepts indicates that the system is packaged by feature.

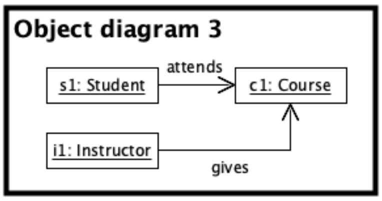
Q14 (1 point): Which of the following object diagrams is consistent with the UML class diagram?

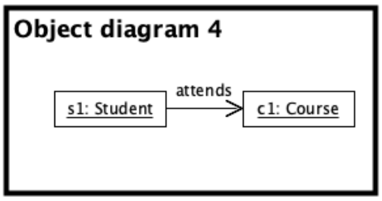
<do not randomize order of options>









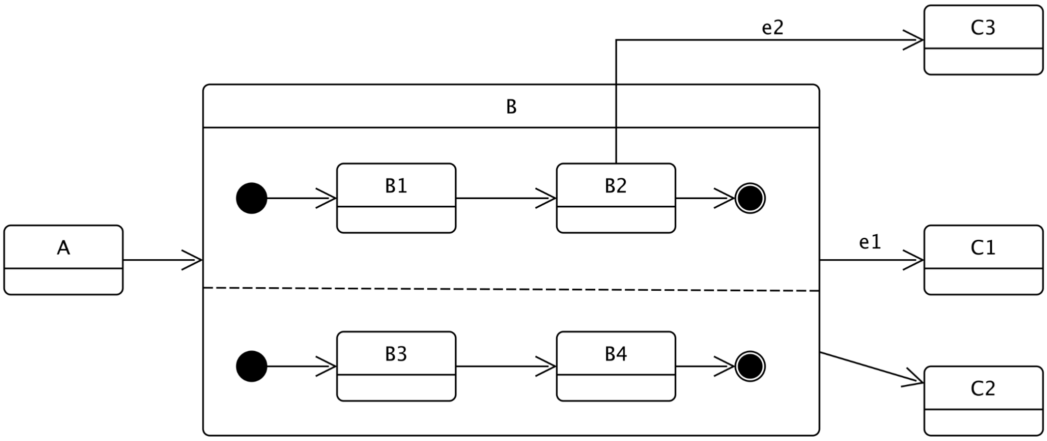


A. Object diagram 1
B. Object diagram 2
C. Object diagram 3
D. Object diagram 4

Q15 (1 point): Which of the following statements does **not** apply to the `loop` fragment of a UML sequence diagram?

A. The loop fragment has only a single operand.
B. If the minimum and maximum numbers of iterations are not specified, the default value is (1, *).
C. A fragment with `loop(5)` and the guard `[x < 5]` is executed exactly 5 times, independently of the guard.
D. The minimum and maximum numbers of iterations can be defined, but they are not mandatory.

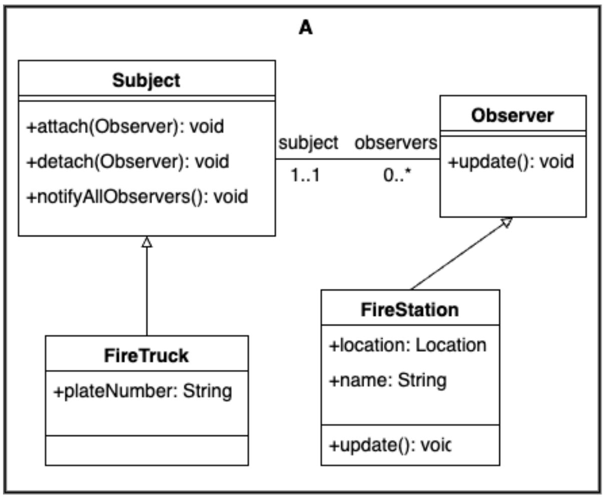
Q16 (1 point): Given the following UML state machine diagram, which of the following statements is **false**?

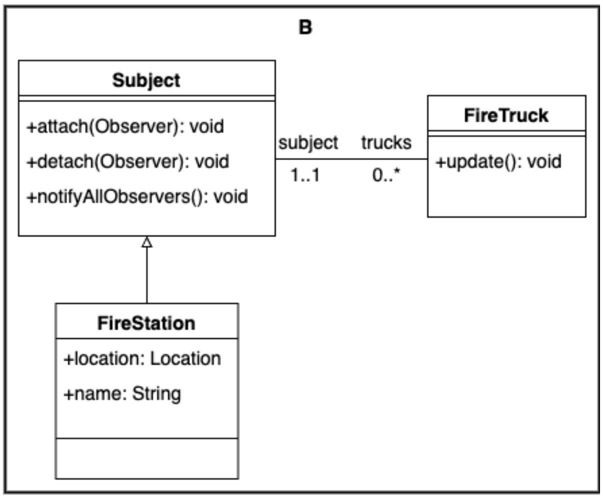


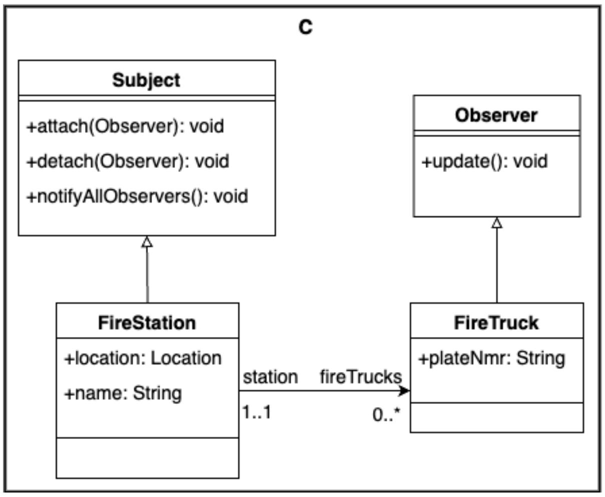
A. State C1 is reached when both the two orthogonal regions of B have reached their final states.
B. When event e2 occurs, B is exited only if B2 is currently active.
C. The only way to reach state B2 is by passing through state B1.
D. When B is active, two of its substates will be active at the same time.

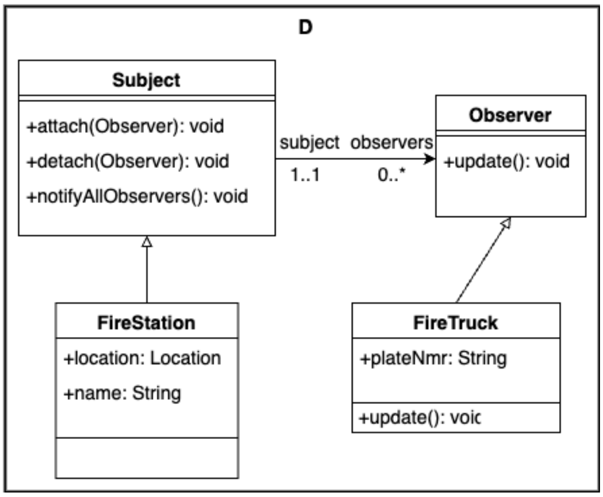
Q17 (1 point): Consider a firefighting system where the central Fire Station monitors multiple fire trucks deployed in different locations. The Fire Station needs to notify all fire trucks whenever there is a new emergency. Which class diagram best represents the implementation of the Observer design pattern in this scenario?

<do not randomize order of options>









A. Solution A
B. Solution B
C. Solution C
D. Solution D

Q18 (1 point): Which design principle is violated by the Java code snippet shown below?

```java
public class OrderClient {
    private MusicShop musicShop;

    public boolean testRetrieval() {
        int numBooklets = this.musicShop.getCollection().getGenre("Rock")
          .getAlbums().find("Master of Puppets").getNumBooklets();
        // ...
    }
}
```

A. Interface segregation principle
B. Law of Demeter
C. Single responsibility principle
D. Liskov substitution principle

Q19 (1 point): Which of the following statements about complexity is **true**?

A. It is more important for a module to have a simple interface than a simple implementation.
B. It is good practice to have the same snippet of code appearing in many parts of your system.
C. One method should perform as many tasks as possible.
D. It is more important for a module to have a simple implementation than a simple interface.

Q20 (1 point): Which of the following statements about Java source code is **true**?

A. Having a class that is mostly accessed through getter and setter methods is always a good design choice.
B. It is advised to reduce pairwise dependencies across classes as much as possible.
C. If you believe that a method will be useful in the future, it is advised to implement it, even if it is never called in the current version of the system.
D. Long call chains of getter methods are an indication of good design.